Projection and IOAPI Metadata

This tutorial focuses on the geographic projection and the IOAPI metadata that is necessary for creating and IOAPI file.

All of the projection and IOAPI metadata is stored together in an object:

```
## get a copy of the ioM object
## this contains all the IOAPI specific metadata
## that is not stored in a regular cdms variable
ioM = o3.ioM.copy()
print ioM
##
         gridName: M_12_IL
         gridType: 2
         ioapi fileType: 1
         projection: Lambert Conformal Conic
         projection parameters:
                 xcent: -90.000000 Degrees E
                 ycent: 40.000000 Degrees N
                 p_alp: 30.000000
                 p_bet: 60.000000
                 p_gam: -90.000000
         verticalType: 2
         vertical top: 10000.000000
         boundary thickness: 1

         nvars: 1
         variable name list: ['O3']

         variable unit list: ['ppmV            ']

         variable desc list: ['Variable O3']

         variable type list: [5,]
         upname: WR_ACONC
         file description: Concentration file output
          Averaged over the synchronization time steps
          Timestamp represents beginning computed date/time
          Layer mapping (CGRID to AGRID):
          Layer  1 to  1

##
```

This variable uses a Lambert Conformal Conic projection, which is described by a center location (−90E, 40N), two parallels (30 and 60 degrees N), and a meridian (−90 degrees E). This projection has a native coordinate system that describes locations in meters from the projection center. I use the terms "xLon" and "yLat" to refer to these native coordinates and to differentiate them from Longitude and Latitude, which are commonly thought of in terms of degrees.

If you think of a specific iovar variable, then you can describe a particular location in three different ways: in terms of row/column, in terms of yLat/xLon (in meters), or in terms of Latitude/Longitude (in degrees). How do they relate? Row and column are simply a 0 based index that indicates which cell you are in. In our o3

example, the dimensions of the array are (24, 1, 70, 58); therefore, the row ranges from 0 to 69 and the column from 0 to 57, where (0,0) is the Southwest cell of the grid. The native coordinates yLat and xLon describe the exact location of a point. In our example data, the seperation between the grid cells is 12,000 meters. It doesn't matter where you are in the domain, the seperation between grid cells is the same. In other words, we have a regular grid in the native coordinate system (yLat and xLon). A grid cell which has the coordinates −6000 yLat and 18000 xLon means that it's center is 6000 m South and 18,000 m East of the projection center. In contrast, o3's grid is irregular if you look at it in terms of Latitude and Longitude. The seperation between adjacent grid cells, in degrees, varies as you move through the domain.

As you start to manipulate and query your iovar variables, it is very important that you keep in mind which system you are using to locate data (see "Subsetting" and "Coordinate Conversion" tutorials for specific examples comparing these systems).

The above print command displays additional information that is needed to write an IOAPI formatted file. To view this metadata in the IOAPI file itself, use ncdump from the unix command prompt:

```
$ ncdump -c ~/tmp/CCTM_ACONC.D2.001 |less
```

Contents Previous Next